Remy Sharp
@rem
Sometimes does flash-ing
1. video & Audio
2. Real-time
3. Graphics
Video killed the radio Flash star?
Who would jump to create another video player?
IE9 & all others supported
<video src="dizzy.mp4"></video>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset=utf-8 />
    <title>Video Example</title>
</head>
<body>
    <video src="dizzy.mp4" controls></video>
</body>
</html>
WATCH WHATEVER WHENEVER.

With Sony’s Betamax SL 4600 video recorder, you can see any TV show you want to see anytime you want to see it. Because Betamax, which plugs into any TV set and is easy to operate, can videotape a show up to three hours long (with the S-100 Videocassette) while you’re doing something else—even while you’re out of the house, by setting the electronic timer. It can also videotape something off one channel while you’re watching another channel.

And remember, Sony has more experience in vidisojectors than anyone else. In fact, we’ve sold more videotapes to broadcasters and industry than any other consumer manufacturer. We even make our own tapes.

For years, you’ve watched TV shows at the times you’ve had to. Now you can watch them at the times you want to.

SONY BETAMAX
THE LEADER IN VIDEO RECORDING
<video controls>
  <source type="video/mp4" src="dizzy.mp4" codecs="avc1.42E01E, mp4a.40.2"/>
  <source type="video/webm" src="dizzy.webm" codecs="vp8, vorbis"/>
  <source type="video/ogg" src="dizzy.ogv" codecs="theora, vorbis"/>
</video>
<video controls>
  <source type="video/mp4" src="dizzy.mp4"/>
  <source type="video/webm" src="dizzy.webm"/>
  <source type="video/ogg" src="dizzy.ogv"/>
</video>
<video controls>
  <source src="dizzy.mp4"/>
  <source src="dizzy.webm"/>
  <source src="dizzy.ogv"/>
</video>
<video controls>
   <source src="dizzy-hd.mp4" media="(min-device-height: 720px)s">
   <source src="dizzy-regular.mp4">
   <...>
</video>
<video width="640" height="360" poster="dizzy.jpg" controls>
  <source src="dizzy.mp4" type="video/mp4" />
  <source src="dizzy.webm" type="video/webm" />
  <source src="dizzy.ogv" type="video/ogg" />
<!--[if gt IE 6]--><object width="640" height="375" classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"><![endif]--><!--[if !IE]><!-->
  <object width="640" height="375" type="video/quicktime" data="dizzy.mp4">
    <!--<![endif]-->
    <param name="src" value="dizzy.mp4" />
    <param name="showlogo" value="false" />
  <object width="640" height="380" type="application/x-shockwave-flash"
    data="player.swf?image=dizzy.jpg&file=dizzy.mp4">
    <param name="movie" value="player.swf?image=dizzy.jpg&file=dizzy.mp4" />
    <img src="dizzy.jpg" width="640" height="360" alt="Title of video"
      title="No video playback capabilities, please download the video below" />
  </object>
<!--[endif]--><![endif]-->
</object>
<!--[if gt IE 6]--></object><!--<![endif]--></video>

<p>Download Video: <a href="dizzy.mp4">High Quality "MP4"</a> | <a href="dizzy.ogv">Low Quality "OGG"</a></p>

http://camendesign.com/code/video_for_everybody
Scripting
var video = document.getElementById('myVideo');
if (video.paused) {
    video.play();
}
```javascript
var video = document.getElementById('myVideo');
if (video.paused) {
    video.play();
}

// position & asTime defined elsewhere
video.addEventListener('timeupdate', function () {
    position.innerHTML = asTime(this.currentTime);
}, false);
```
Simple API

Methods: play(), pause(), canPlayType(mime)

Properties: currentTime, paused, duration, loop, autoplay, muted, volume, etc

Events: loadedmetadata, canplay, progress, play, pause, seeking, timeupdate, etc
Warning! User agents should not provide a public API to cause videos to be shown full-screen. A script, combined with a carefully crafted video file, could trick the user into thinking a system-modal dialog had been shown, and prompt the user for a password. There is also the danger of "mere" annoyance, with pages launching full-screen videos when links are clicked or pages navigated. Instead, user-agent specific interface features may be provided to easily allow the user to obtain a full-screen playback mode.
Video

Note that (at time of writing) Webkit nightly supports full screen mode, which will add a button above.

HTML5 demo
Maximising the serendipity of travel
Matt Biddulph, CTO of Dopplr

This is a proof of concept demonstration of HTML pluginless Video and Audio playback, showing AV content synchronised to chapters, allowing direct access to specific sections.
Yay!

1. No plugins required
2. Simple API: play, pause, etc
3. Video & Audio: the same
4. HTML & CSS - no compile or different skills required
1. Codecs
2. Our friend IE
Flash will be needed as a backup to video for a while yet.
This collaborative drawing application was built during the HTML5 APIs workshop with Remy Sharp during Web Directions @media 2010. Clicking images in the My Captures gallery will open up a new window for each individual drawing.
WebSocket

- Low latency
- Bi-directional
- No same-original rules
- Chrome, Safari, MobileSafari & Firefox
- Fallback on Flash...ironically
var ws = new WebSocket("ws://myserver.com/");

ws.onmessage = function(event) {
    var data = JSON.parse(event.data);
};

ws.onclose = function() {
};

ws.onopen = function() {
};
ws.onmessage = function (event) {
    var data = JSON.parse(event.data);
};

All message data lives here
EventSource

- Server pushed events
- Same-original rules apply
- Can fallback with JavaScript
```javascript
var es = new EventSource("/x-service/");

es.onmessage = function (event) {
  var data = JSON.parse(event.data);
};

es.onopen = function () {};
```
Graphics
2D Graphics
Canvas API
```javascript
var canvas = document.getElementsByTagName('canvas')[0],
    ctx = canvas.getContext('2d');
```
canvas.getContext(‘2d’)
Google Analytics – Flash charts

Interactive Demo – Canvas charts
A Tale Of Two Cities
function canvas(w, h) {
    var ctx = document.createElement('canvas').getContext('2d'),
        canvas = ctx.canvas;
    canvas.width = w;
    canvas.height = h;
    return ctx;
}

var rainbow = canvas(100, 1),
    rainbowGrad = createRainbow(rainbow);

rainbow.fillStyle = rainbowGrad;
var imageData = rainbow.getImageData(0, 0, 100, 1),
    pixels = imageData.data;

// loop over each pixel and create the dot image
for (var i = 0; i < pixels.length; i += 4) {
    createPoint(pixels, i);
}
function createPoint(pixels, i) {
    var dot = canvas(24, 24);

    // outer white circle
    dot.fillStyle = '#fff';
    dot.arc(12, 12, 10, 0, Math.PI * 2, true);

    // drop shadow
    dot.shadowBlur = 2;
    dot.shadowColor = 'rgba(0,0,0,.7)';
    dot.shadowOffsetX = 2;
    dot.shadowOffsetY = 2;

    // fill outer ring
    dot.fill();

    // remove shadow
    dot.shadowBlur = 0;
    dot.shadowColor = 'rgba(0,0,0,0)';
    dot.fillStyle = 'rgb(' + [pixels[i],   // red
        pixels[i+1], // green
        pixels[i+2]  // blue
    ].join(',') + ')';

    // start inner circle
    dot.beginPath();
    dot.arc(12, 12, 8, 0, Math.PI*2, true);

    // fill inner circle
    dot.fill();

    new google.maps.MarkerImage(
        dot.canvas.toDataURL('image/png')
    );
}
new google.maps.MarkerImage(
    dot.canvas.toDataURL('image/png')
);
1. Timer paints video into canvas
2. Reads all pixels for bright spots
3. Translates to the vector
4. Draws selected input

1. Flash and canvas share the same black box features

2. People will abuse the technology
Scalable Vector Graphics
Native SVG in Firefox...
...using Flash!
Yeah, 3D CSS.
```css
#canvas {
  -webkit-perspective: 800;
  -webkit-perspective-origin: 50% 20em;
}

#rubiks {
  -webkit-transform-style: preserve-3d;
  -webkit-transform: rotateX(15deg) rotateY(15deg);
}

#rubiks .face1 {
  -webkit-transform: rotateX(90deg) translateZ(10.8em);
}

#rubiks .face2 {
  /* front */
  -webkit-transform: translateZ(10.8em);
}

#rubiks .face3 {
  -webkit-transform: rotateY(90deg) translateZ(10.8em);
}

#rubiks .face4 {
  /* back face */
  -webkit-transform: rotateY(180deg) translateZ(10.8em);
}

#rubiks .face5 {
  -webkit-transform: rotateY(-90deg) translateZ(10.8em);
}

#rubiks .face6 {
  -webkit-transform: rotateX(-90deg) translateZ(10.8em) rotate(180deg);
}
```
WebGL
AKEMI

You see a demonstration and proof-of-concept of how smoothly WebGL and Haxe combine to a nice web experience by shading and composing the famous strange-attractor.

START
Mobile?
WebGL in Firefox on Nokia N900
#tweetcoding with canvas

**Imrivas**

Oscilloscope: Noisy pulse

http://twitter.com/Imrivas/status/1267192483

Mon, 02 Mar 2009 08:06:55 +0000

```
i++;g.clear();$\text{l}();\text{for}(j=0;j<600;j++)\{\text{g.drawCircle}(j,200+50*\text{sin}(\pi * j / 2)-25*\text{cos}(\pi * j / 5)),1}\}
```

See Result

See submission tweet

Mon Mar 02 00:06:19 +0000 2009

---

**Imrivas**

#tweetcoding "diagonal snake" use the mouse to direct the snake and go as long as you can.

http://twitter.com/tomee6/status/1267109791

Sun, 01 Mar 2009 23:41:19 +0000

```
if(i++)$\text{g.clear}();\text{mi}(5,5)=\{\text{x:5,y:5}\};a='+'o.x+o.y;b=0[a]\text{ds(b:b'=42,7<<b});b?i=0:o[a]=23j(o.x==o.x<\text{mouseX}\text{}?-5,0.y==o.y<\text{mouseY}\text{}?-5);
```

See Result

See submission tweet

Sun Mar 01 23:39:25 +0000 2009

---

**LorenBednar**

#tweetcoding "One Sine Day" mouse left and right to move :) (fixed)

http://twitter.com/LorenBednar/status/1266830638

Sun, 01 Mar 2009 22:13:20 +0000
...or just ask a question :)